

REMARKS

Claims 1-54 are pending in the case. Claims 8-11 are currently amended. Claim 54 is new. Claims 28-33 are allowed. Claims 20, 42, and 43 are objected to.

Interview Summary

An interview was conducted on October 30, 2009 with an attorney of record, Jeffery Pyle, and the Examiner to discuss claim 1. Claim 1 and the cited art were generally discussed. No final agreement was reached.

Allowable Subject Matter

Applicants acknowledge and appreciate that claims 28-33 are allowed.

The Examiner objected to claim 20 as being dependent upon a rejected claim 18. The Examiner indicated that claim 20 would be allowable is rewritten in independent form. Claims 42 and 43 are objected to as being independent upon a rejected claim 38, but would be allowable if rewritten in independent form.

Claims Rejections Under 35 USC §102

The Examiner rejected claims 1-4, 6-12, 15-17, 19, 21-24, 27 38-41, 44-46 and 48-53 as being anticipated under 35 U.S.C. §102(b) by U.S. Patent 5,204,960 (“*Smith*”). To anticipate, *Smith* must disclose every limitation of the rejected claim in the same relationship to one another as set forth in the claim. M.P.E.P. § 2131; *In re Bond*, 15 U.S.P.Q.2d (BNA) 1566, 1567 (Fed. Cir. 1990). *Smith* fails to do this if it is construed properly.

For ease of illustration, claim 1 is discussed first. Claim 1 recites “[a] method for use in developing a program, comprising compiling at least a portion of a source code program defined by a waypoint during the editing of the source code program.” By allowing speculative compilation during the editing of a source file, a programmer may be able to reduce development

and debug time associated with software code. It should be noted that examples illustrated herein are exemplary in nature only, and do not limit the claims.

The Examiner's rejection under §102 is not proper at least because *Smith* fails to teach at least one claimed feature. For example, claim 1 calls for compiling at least a portion of a source code program defined by a waypoint. The Specification is instructive as to the definition of a waypoint, as called for in claim 1: "A 'waypoint' is a point in the source code program 100 defining a bound of a portion of the source code program 100 for which it is desirable to begin compiling prior to the end of an editing session." Specification, p.9, ll. 2-4 (*emphasis added*). The Examiner argues that *Smith* anticipates the claimed feature of a waypoint because *Smith* teaches boundaries that can be defined in source code files. See Office Action dated Nov. 30, 2009, p.2 (citing column 2 of *Smith*). *Smith* teaches that boundaries may be set in source code files that will correspond to respective portions of machine code in object files after compilation of the source code. See *Smith*, col. 2, ll. 4-55. As such, *Smith* does not, and cannot, teach a waypoint, as called for in claim 1. Applicants respectfully submit that references to the Specification are provided for illustrative purposes only and do not operate to restrict or narrow any portion of the claims.

Similarly, just as *Smith* does not teach a waypoint, as called for in claim 1, *Smith* does not, and cannot, teach "compiling at least a portion of a source code program defined by a waypoint during the editing of the source code program" (*emphasis added*). In other words, compilation of at least a portion of the source code may occur while the source code is being edited, not simply after the editing has been completed (as taught in *Smith*). The Examiner argues, as described above, that *Smith* teaches boundaries that can be defined in source code files. See Office Action dated Nov. 30, 2009, p.2 (citing column 2 of *Smith*). *Smith* allows for

recompilation for certain parts of source code, and thus corresponding parts of object files, if it is determined that the portions of the source code and portions of the corresponding object file do not match up; however, the recompilation only occurs *after* the source code has been edited, not during at least a portion of a time period in which the source code is edited. In contrast, claim 1 recites “compiling at least a portion of a source code program defined by a waypoint during the editing of the source code program” (*emphasis added*).

For at least the aforementioned reasons, claim 1 and its dependent claims are allowable. For at least similar reasons, the remaining claims are also allowable.

Other claims are also allowable for additional features recited therein. For example, claim 12 recites “modifying the identified file reader to read a portion of a source code program defined by a waypoint from a standard input.” The Examiner argues that *Smith* teaches this claimed feature because *Smith* discloses “a user edits a file.” See Current Office Action, p.6 (citing *Smith*, col. 3, ll. 52-58 and Figure 1) (*emphasis in Office Action*). Applicants would like to respectfully point out that the Examiner is referring to editing a file (*i.e.*, editing a source code file), while claim 12 specifically recites “modifying the identified file reader.” The file reader is a part of the compiler which *compiles the source code*. See Specification, ¶[0010]. As such, editing a file, which is part of the source code, is clearly not the same as “modifying the identified file reader,” which is part of a compiler.

For at least the aforementioned reasons, claim 12 and its dependent claims are allowable.

Claims Rejections Under 35 USC §103

The Examiner rejected claims 5, 18 and 47 under 35 U.S.C. 103(a) as being unpatentable over *Smith*, in view of US Patent No. 6,836,884 (*Evans*). Applicants respectfully traverse this rejection.

Claim 5 is discussed first. Claim 5 calls for the method of claim 1, further comprising completing editing of the source code and compiling the source code from the second waypoint to the end of the source code.

The Examiner's rejection is not proper at least because *Smith* and *Evans*, either alone or in combination, fail to teach or suggest all the features of claim 5. For example, claim 5, similarly to claim 1, recites a waypoint. *Smith* fails to teach certain claim limitations as shown above with respect to the Examiner's §102 rejection, and *Evans* fails to remedy the deficiencies of *Smith*. *Evans*, like *Smith*, is concerned with compilation *after* editing, not during editing as called for in claim 1.

For at least these reason, claim 5 is allowable. For at least similar reasons, claims 18 and 47, along with their respective dependent claims, are also allowable.

The Examiner rejected claims 13, 14, 25, 26 and 34-37 under 35 U.S.C. 103(a) as being unpatentable over *Smith*, in view of U.S. Pub. No. 20020087916 (*Meth*). Applicants respectfully traverse this rejection.

The Examiner's rejection is not proper at least because *Smith* and *Meth*, either alone or in combination, fail to teach or suggest all the features of claim 5. For example, claim 5, similarly to claim 1, recites a waypoint. *Smith* fails to teach certain claim limitations as shown above with respect to the Examiner's §102 rejection, and *Meth* fails to remedy the deficiencies of *Smith*. *Meth*, like *Smith*, is not concerned with compilation *during* editing. Indeed, *Meth* is not concerned with compiling; rather, *Meth* is concerned with "checkpointing" processes at an application layer. In other words, *Meth* teaches that an application may check point certain steps in its execution. In contrast, claim 1 recites a waypoint, as defined in the Specification. As such, a combination of *Smith* and *Meth* does not, and cannot, teach all the features of claim 13.

Further, without using improper hindsight reasoning and using the claim as a roadmap, the person of ordinary skill in the art would have no apparent reason to modify the references to arrive at the subject matter of claim 13. The Examiner essentially provided a conclusory statement that adding the features of these references together would make for a better product; *i.e.*, the Examiner has simply stated the result of such a combination. *See* Final Office Action, p.4. The Examiner states:

“[I]t would have been obvious to a person of ordinary skill in the art at the time the invention was made to modify teaching of Evans with the teachings of Meth to include modifying the identified file reader to read from the standard input includes modifying the identified file reader to read from an open system call in order to use to open and close files whenever the program opens a file with the open system call, the user-level checkpoint mechanism intercepts the open system call and records for itself the name of the file being open (see [0039])...”. *Id.*

The Examiner’s reasoning is conclusory, if not unclear. In other words, the Examiner has merely stated that such a combination would have been obvious because of the result of the combination. This reasoning is circular and improper. The Examiner has not pointed to any teachings in the cited references that would **motivate** a person of skill in the art to combine the references. In other words, the question that must be addressed includes “*why* would a person have thought to combine the cited references based on their teachings?”, and “*what* was the need?”, not simply “what benefits would result?”. Motivation to combine aside, as discussed above, even if *Smith* and *Meth* were to be combined, claim 13 as a whole would be untaught and non-obvious over the references.

For at least these reason, claim 13 is allowable. For at least similar reasons, claims 14, 25, 26 and 34-37, along with any respective dependent claims, are also allowable.

Applicants therefore respectfully submit that the claims are in condition for allowance, and requests that they be allowed to issue. The Examiner is invited to contact the undersigned

attorney at (713) 934-4069 with any questions, comments or suggestions relating to the referenced patent application.

Respectfully submitted,

WILLIAMS, MORGAN & AMERSON, P.C.

Date: March 1, 2010

By: /Jaision C. John/
Jaision C. John, Reg. No. 50,737
10333 Richmond, Suite 1100
Houston, Texas 77042
(713) 934-4069
(713) 934-7011 (facsimile)
ATTORNEY FOR APPLICANT(S)